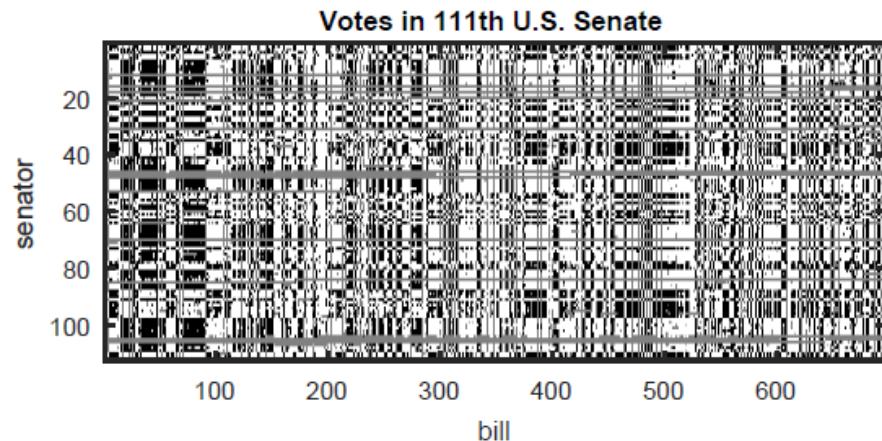


# Chapter 7

## Matrix Analysis



## Section 7.1

# From matrix to insight

# From matrix to insight

- Any two-dimensional array of numbers may be interpreted as a matrix
- These may come from widely disparate tasks
- Examples:
  - Text/document search
  - Voting patterns
  - Preferences/ratings *Netflix recommendations*
  - Graphs
  - Networks: social, political, co-authorship, casting in movies,...
  - Images

## Examples of matrices

- A *term-document matrix* may be used for analyzing a body of documents (or *corpus*)
- Each column may be a document; each row a term
- E.g, your textbook may have words like “numerical,” “discretization,” “matrix,” “integration” and “function”
- An analysis textbook may have words like “integration,” “function,” “continuous” and so forth
- The occurrence of “function” may be often in both books, but the other terms are likely to be much different in frequency
- Meaning could be inferred from this kind of approach: latent semantic analysis

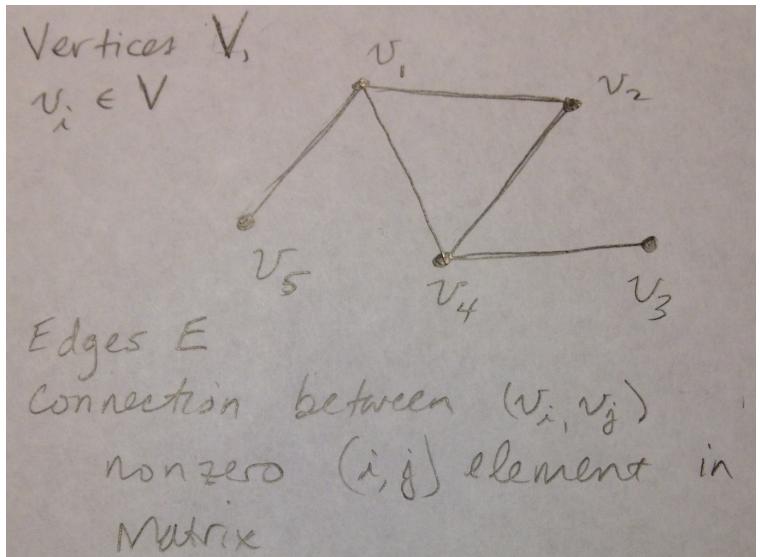
## Examples of matrices

- A *term-document matrix* example: matrix in green

Term	FNC by TAD+RJB	Analysis by Rudin	SM by Trefethen	NYT coffee table book
Numerical	251	2	179	0
Integration	37	275	33	18
Function	175	345	123	0
Matrix	151	11	87	0
Continuous	15	212	11	0
Spectral	15	0	124	0
Citizen	0	0	0	13

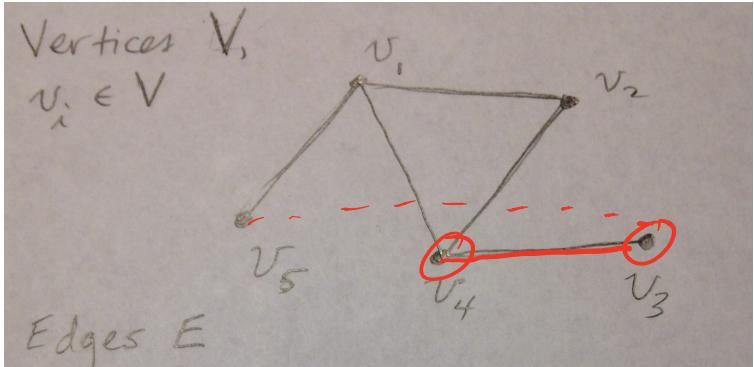
# Graphs as matrices

- Could be social network, internet (or subset of it), the web, etc
- Consider graphs first
- A graph is a set of nodes  $V$  connected by set of edges  $E$
- Sometimes the graph is denoted  $G(V, E)$



# Adjacency matrix

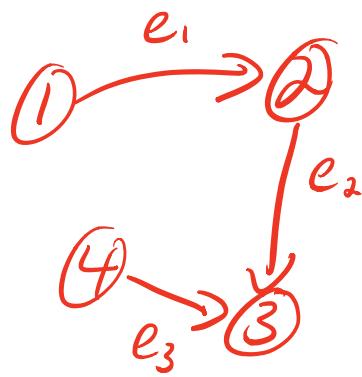
- We want adjacency matrix  $A = \{a_{ij}\}$  that represents this graph
- Edges are unweighted and undirected
- If an edge between  $(v_i, v_j)$ , then a one is placed in both  $a_{ij}$  and  $a_{ji}$  (symmetric)
- No self-connections
- Corresponding matrix at right



	1	2	3	4	5
1	0	1	0	1	1
2	1	0	0	1	0
3	0	0	0	1	0
4	1	1	1	0	0
5	1	0	0	0	0

## Incidence matrix:

$$\begin{matrix} & e_1 & e_2 & e_3 \\ 1 & -1 & 0 & 0 \\ 2 & 1 & -1 & 0 \\ 3 & 0 & 1 & 1 \\ 4 & 0 & 0 & -1 \end{matrix}$$



## Adjacency matrix: Buckyball

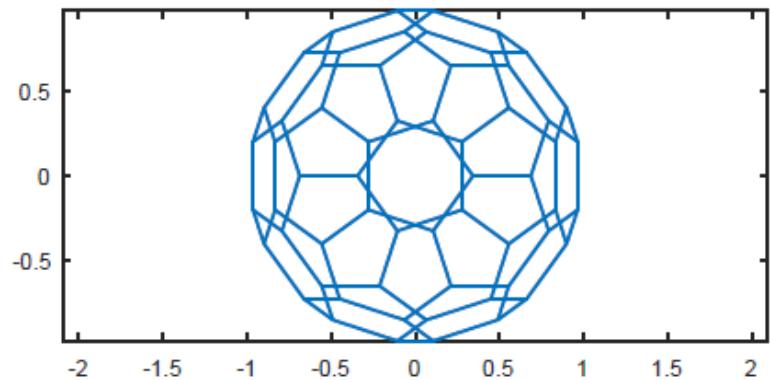
- Matlab has a built-in example of a graph representing the arrangements of carbon atoms in a  $C_{60}$  molecule, a.k.a. the buckyball:

```
[A,v] = bucky;  
size(A)
```

```
ans =  
60    60
```

- The output has the adjacency matrix A and the vertex locations v
- Plotting the graph:

```
gplot(A,v), axis equal
```

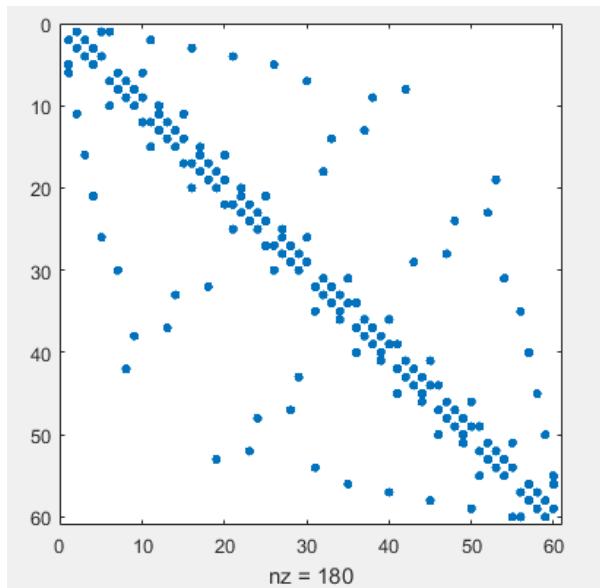


## Adjacency matrix: Buckyball

- We know there are 60 nodes; how many edges?
- There is more than one way to compute this
- For undirected, you could use the spy command:

```
>> [A, v]=bucky;  
>> spy(A)  
>>
```

- Is nnz it? Not quite
- You could use the triu and sum commands; how?



[Example 7.1.4]

# Images

- Pictures and images are matrices in matlab
- Use `imread` and `imshow` to display them

```
A = imread('peppers.png');  
size(A)
```

```
ans =  
    384    512     3
```

- Three “layers” are RGB components
- To display: `imshow(A)`



## Images

- We can convert to grayscale to get only a 2D matrix
- Use `rgb2gray` and `double`:

```
A = rgb2gray(A); % collapse from 3 dimensions to 2  
A = double(A); % convert to floating point  
[m,n] = size(A)
```

```
m =  
    384  
n =  
    512
```

- To display: `imshow(A, [0, 255])`
- What if different range spec'd?



[Example 7.1.5]

## Section 7.2

# Eigenvalue decomposition

# The eigenvalue decomposition

- We can decompose a matrix  $A$ , under some assumptions, into a useful set of other matrices using its eigenvalues and eigenvectors.
- Recall that the eigenvalue problem is  $Ax = \lambda x$ , for the eigenvalues  $\lambda$  and eigenvectors  $x$   $x \neq 0$
- (Sometimes the eigenvectors are referred to as the eigenspaces.)
- We need to review some results and terminology from linear algebra before proceeding
- In particular we need to recall the complex-valued case

$$S_\lambda = \{x \in \mathbb{C}^n \mid Ax = \lambda x\} \leftarrow \text{eigenspace}$$

## Complex vectors and matrices

- Recall complex numbers  $x = a + ib$ , where  $i^2 = -1$  and  $a, b$  are real-valued.
- The complex conjugate is  $\bar{x} = a - ib$
- For a matrix  $A$  with complex-valued elements, the transpose needs to use complex conjugation at the same time:  $A^* = (\bar{A})^T = \overline{A^T}$
- The hermitian will keep certain properties analogous with the real-valued case
- e.g.,  $(A^T A)^T = A^T A$  so that  $A^T A$  is symmetric. So is  $(A^* A)^* = A^* A$
- Essentially, hermitian replaces transpose for complex-valued matrices and vectors

## Complex vectors and matrices

- Essentially, hermitian replaces transpose for complex-valued matrices and vectors
- For inner products with complex  $\mathbf{u}$  and  $\mathbf{v}$ :  $\mathbf{u}^* \mathbf{v} = \sum_{k=1}^n \bar{u}_k v_k,$
- This defines the two-norm for vectors,  $\|\mathbf{u}\|_2 = \sqrt{\mathbf{u}^* \mathbf{u}}$ , and this will in turn define the two norm for matrices
- Definition of orthogonal
- And orthonormal set of vectors  $\mathbf{u}_j, j = 1, 2, \dots, n$ :  $\mathbf{u}_i^* \mathbf{u}_j = 1$  if  $i = j$ , otherwise  $\mathbf{u}_i^* \mathbf{u}_j = 0$

$$\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_n]$$

$$\mathbf{U}^* \mathbf{U} = \mathbf{I}$$

## Complex vectors and matrices

- A square **real-valued matrix** with orthonormal columns is an orthogonal matrix
- A square **complex matrix** with orthonormal columns is unitary
- An  $n \times n$  unitary matrix  $\mathbf{U}$  satisfies  $\mathbf{U}^{-1} = \mathbf{U}^*$  and  $\|\mathbf{U}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$  for any complex vector  $\mathbf{x} \in \mathbb{C}^n$

$$\mathbf{U}^* \mathbf{U} = \mathbf{I}$$

# The eigenvalue problem

- The eigenvalue problem is  $Ax = \lambda x$ , for the eigenvalues  $\lambda$  and eigenvectors  $x$
- It can also be written  $0 = (\lambda I - A)x$
- We find eigenvalues  $\lambda_k$  such that  $\lambda_k I - A$  is singular, and we find the corresponding eigenvectors  $x_k$  for  $k = 1, 2, \dots, n$
- When doing this by hand, one calculates the roots of the characteristic polynomial  $\det(\lambda I - A)$
- There are thus  $n$  eigenvalues for an  $n \times n$  matrix, counting multiplicity
- Note: eigenvalues and eigenvectors not done that way in computer

# The eigenvalue decomposition

- Suppose we know the eigenvalues  $\lambda_k$  and eigenvectors  $\mathbf{v}_k$  so that  $A\mathbf{v}_k = \lambda_k \mathbf{v}_k$ , for  $k = 1, 2, \dots, n$
- For each  $k$ , we have a column vector on each side of the equation
- We can assemble each of those  $n$  column vectors into a matrix:

$$\rightarrow [A\mathbf{v}_1 \quad A\mathbf{v}_2 \quad \cdots \quad A\mathbf{v}_n] = [\lambda_1 \mathbf{v}_1 \quad \lambda_2 \mathbf{v}_2 \quad \cdots \quad \lambda_n \mathbf{v}_n]$$

• Rewrite:  $\rightarrow A [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n] = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n] \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$

• Matrix form:  $\rightarrow \mathbf{AV} = \mathbf{VD}$ . (7.2.3)

# The eigenvalue decomposition

- If we know that there is a complete set of linearly independent (LI) eigenvectors  $v_k$ , then  $V^{-1}$  exists, and we can postmultiply by it to get:

$$A = VDV^{-1}$$

- This is the eigenvalue decomposition, or EVD, of  $A$
- If  $A$  is diagonalizable, it will have an EVD
- When does that happen?

## Theorem 7.2.1

If the  $n \times n$  matrix  $A$  has  $n$  distinct eigenvalues, then  $A$  is diagonalizable.

## Computing EVD in Matlab

- We can use the `eig` command to compute the EVD
- The eigenvalues arrive as the diagonal elements of  $D$
- The eigenvectors are the columns of  $V$
- $A$  is singular, but has distinct eigenvalues, and because of that, it has a complete set of LI eigenvectors.  
Because of that,  $V$  is nonsingular

```
A = pi*ones(2, 2);  
lambda = eig(A)
```

```
lambda =  
    0  
  6.2832
```

```
[V, D] = eig(A)
```

```
V =  
    -0.7071    0.7071  
    0.7071    0.7071  
D =  
    0         0  
    0     6.2832
```

## Computing EVD in Matlab

- We can easily check that the EVD is equivalent to A:

```
>> norm( A-V*D/V ) % /V is equivalent to *inv(V)  
  
ans =  
  
8.8818e-16  
  
>> |
```

---

- What if A is not diagonalizable?

## Computing EVD in Matlab

- If  $A$  is not diagonalizable, it will not have all eigenvalues distinct, and it will not have a full set of eigenvectors
- `eig` still works, but  $V^{-1}$  doesn't exist:
- $V$  is only rank 1, indicating a single LI column
- You should get in the habit of critically evaluating output to check whether it is consistent with theory

[Example 7.2.1]

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

```
[V, D] = eig([1 1; 0 1])
rankV = rank(V)
```

```
V =
    1.0000    -1.0000
            0     0.0000
D =
    1         0
    0         1
rankV =
    1
```

## The EVD and similarity

$$A = VDV^{-1}$$

- There is an important relationship between  $A$  and  $D$
- For any nonsingular matrix  $S$ , then  $B = SAS^{-1}$  is said to be similar to  $\underline{A}$
- In many basic linear algebra texts, one can find a proof of this result:

Theorem 7.2.2

If  $X$  is an nonsingular matrix, then  $XAX^{-1}$  has the same eigenvalues as  $A$ .

- There is a really nice interpretation of the similarity transformation

$$\begin{aligned} \det(\lambda I - XAX^{-1}) &= \det(\lambda XX^{-1} - XAX^{-1}) \\ &= \det(X(\lambda I - A)X^{-1}) \\ &= \cancel{\det(X)} \det(\lambda I - A) \cancel{\det(X^{-1})} \\ &= \det(\lambda I - A). \end{aligned}$$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = 1 \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 2 \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 3 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

## Similarity transformations

• Consider the ~~product~~ of a nonsingular matrix  $X$  with any vector:

$$\mathbf{y} = X\mathbf{z} = z_1\mathbf{x}_1 + \cdots + z_n\mathbf{x}_n$$

- The  $\mathbf{x}_i$  are the columns of  $X$
- The columns of  $X$  are LI because it is invertible, so it is a basis for  $\mathbb{C}^n$
- The  $z_i$  are the *coordinates* of  $\mathbf{y}$  using the columns of  $X$  as a basis
- But, we could left multiply by  $X^{-1}$  and then  $\mathbf{z} = X^{-1}\mathbf{y}$
- The elements of  $\mathbf{y}$  are now coordinates for  $\mathbf{z}$  using the columns of  $X^{-1}$  as a basis
- Thus: *multiplication by the inverse of a matrix performs a change of basis into the coordinates associated with the matrix*

$$[\mathbf{y}]_X = \mathbf{z} = X^{-1}\mathbf{y}$$

$$[\mathbf{z}]_{X^{-1}} = \mathbf{y} = X\mathbf{z}$$

## Similarity transformations

$$V^{-1}x = [x]_V$$

- Now consider the EVD
- Let  $\mathbf{u} = \mathbf{A}\mathbf{x}$ , or

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$$

$$\mathbf{u} = \mathbf{A}\mathbf{x} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}\mathbf{x}$$

- Premultiply by  $V^{-1}$  to get

$$V^{-1}\mathbf{u} = \mathbf{D}(V^{-1}\mathbf{x})$$

$$[\mathbf{u}]_V = \mathbf{D}[\mathbf{x}]_V$$

- This equation says that using the columns of  $V$  for a basis, that there is a diagonal relation between the two vectors  $\mathbf{u}$  and  $\mathbf{x}$
- Said another way, the EVD finds a basis for  $\mathbb{C}^n$  so that the map is diagonal:

$$\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$$

- In that case, each coordinate is just multiplied by its own scalar

## Similarity transformations

- This can be really convenient for matrix powers; these will be an important operation for us
- Consider  $\mathbf{A}^2$ , and use the EVD:

$$\mathbf{A}^2 = (\mathbf{V}\mathbf{D}\mathbf{V}^{-1})(\mathbf{V}\mathbf{D}\mathbf{V}^{-1}) = \mathbf{V}\mathbf{D}(\mathbf{V}^{-1}\mathbf{V})\mathbf{D}\mathbf{V}^{-1} = \mathbf{V}\mathbf{D}^2\mathbf{V}^{-1},$$

- If we ~~knew~~ the EVD, we could just square the diagonal elements of  $\mathbf{D}$ , then reconstruct  $\mathbf{A}^2$
- Higher powers? Then

$$\mathbf{A}^k = \mathbf{V}\mathbf{D}^k\mathbf{V}^{-1}$$

- Raise each eigenvalue to the  $k$ -th power to get  $\mathbf{D}^k$ !

## Conditioning of EVD computation

- There are theorems around to tell us how much eigenvalues change in response to changes in the matrix
- One is the Bauer-Fike theorem:

### Theorem 7.2.3

Let  $\mathbf{A} \in \mathbb{C}^{n \times n}$  be diagonalizable,  $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$ , with eigenvalues  $\lambda_1, \dots, \lambda_n$ . If  $\mu$  is an eigenvalue of  $\mathbf{A} + \mathbf{E}$  for a complex matrix  $\mathbf{E}$ , then

$$\min_{j=1, \dots, n} |\mu - \lambda_j| \leq \kappa(\mathbf{V}) \|\mathbf{E}\|, \quad (7.2.6)$$

where  $\|\cdot\|$  and  $\kappa$  are in the 2-norm.

*not computable*

- Eigenvalues by as much as a factor of the condition number  $\kappa(\mathbf{V})$
- If  $\mathbf{V}$  is unitary,  $\kappa(\mathbf{V}) = 1$ ,  $\mathbf{A}$  is *normal*, eigenvalue calculation robust
- If  $\mathbf{V}$  nearly singular,  $\kappa(\mathbf{V}) \gg 1$ , eigenvalues can change significantly

From Wikipedia "Bauer-Fike theorem":

Alternate formulation:

Let  $(\lambda^a, v^a)$  be an approximate eigenvalue - eigenvector pair and

$$r = Av^a - \lambda^a v^a.$$

Then there is an eigenvalue  $\lambda$  of  $A$  such that

$$|\lambda - \lambda^a| \leq \kappa(V) \frac{\|r\|}{\|v^a\|}.$$

computable  
upper  
bound

# Conditioning of EVD computation

- Example: triangular matrix with  $n = 15$
- The bound on the change in the condition number is around  $1e7$
- Test the theorem with  $1e-7$  size perturbations
- The max change in these few cases is about 25% to 50% of the possible change

[Example 7.2.2]

```
lambda = (1:n)';
A = triu( ones(n,1)*lambda' );
```

The Bauer-Fike theorem provides an upper bound these eigenvalues.

```
[V,D] = eig(A);
kappa = cond(V)
```

```
kappa =
7.1978e+07
```

```
for k = 1:3
E = randn(n); E = 1e-7*E/norm(E);
mu = eig(A+E);
max_change = norm( sort(mu)-lambda, inf )
end
```

```
max_change =
0.2407
max_change =
0.4492
max_change =
0.2737
```

## Method of EVD computation

- The practical methods for computing the EVD are beyond the scope of this class (and book)
  - It is worth pointing out that the methods often use the idea of matrix powers as part of the computation.
  - If the eigenvalues are distinct, then raising them to a power separates them for large powers  $k$
  - There is an easy and elegant way to accomplish this separation
- Power method : e.g PageRank of Google.*

# Method of EVD computation

- Example: create a matrix with known eigenvalues:

```
D = diag([-6 -1 2 4 5]);  
[V, R] = qr(randn(5));  
A = V*D*V'; % note that V' = inv(V)
```

The qr function takes a random 5 by 5 matrix, then returns orthogonal matrix V and upper triangular matrix R

A and D are similar

- Now do QR factorization and reverse it:

$$A = QR$$
$$Q^T A = R$$

```
[Q, R] = qr(A);  
A = R*Q;
```

- We have same eigenvalues!

$$RQ = Q^T A Q \leftarrow \text{similar to } A$$

eig(A)

```
ans =  
-6.0000  
-1.0000  
5.0000  
4.0000  
2.0000
```

## Method of EVD computation

- It turns out that we can repeat this and not change the eigenvalues!
- This is *Francis QR iteration*
- For this example:
- Look at diagonal elements, and off diagonal elements are getting small

```
for k = 1:15
    [Q, R] = qr(A);
    A = R*Q;
end
A
```

A =	-5.9984	-0.1336	0.0100	-0.0000	0.0000
	-0.1336	4.9960	-0.0491	0.0000	-0.0000
	0.0100	-0.0491	4.0024	-0.0001	-0.0000
	-0.0000	0.0000	-0.0001	2.0000	-0.0001
	0.0000	-0.0000	-0.0000	-0.0001	-1.0000

[Example 7.2.3]

## Section 7.3

# Singular value decomposition

## Singular value decomposition

- Here is another matrix factorization; it is widely used in many fields:

Theorem 7.3.1

Let  $\mathbf{A} \in \mathbb{C}^{m \times n}$ . Then  $\mathbf{A}$  can be written as

$$\mathbf{A}\mathbf{V}=\mathbf{U}\mathbf{S}$$

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^*, \quad (7.3.1)$$

where  $\mathbf{U} \in \mathbb{C}^{m \times m}$  and  $\mathbf{V} \in \mathbb{C}^{n \times n}$  are unitary and  $\mathbf{S} \in \mathbb{R}^{m \times n}$  is real and diagonal with nonnegative entries. If  $\mathbf{A}$  is real, then so are  $\mathbf{U}$  and  $\mathbf{V}$  (which are then orthogonal matrices).

- This is the singular value decomposition, or SVD
- Cols of  $\mathbf{U}$ : left singular vectors; cols of  $\mathbf{V}$ : right singular vectors
- Diagonal elements of  $\mathbf{S}$  are the singular values  $\sigma_1, \dots, \sigma_r, r = \min\{m, n\}$

## Singular value decomposition

- The SVD is

$$A = USV^*$$

- $A \in \mathbb{C}^{m \times n}$  is  $m \times n$ , with complex entries
- Columns of (unitary)  $U \in \mathbb{C}^{m \times m}$ : left singular vectors
- Columns of (unitary)  $V \in \mathbb{C}^{n \times n}$ : right singular vectors
- Diagonal elements of  $S \in \mathbb{R}^{m \times n}$  are the singular values

$$\sigma_1, \dots, \sigma_r, r = \min\{m, n\}$$

- Usually the ordering of the singular values is

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$$

- $\sigma_1$  is the principal singular value
- $u_1$  and  $v_1$  are the principal singular vectors

$$Av_i = \sigma_i u_i$$
$$i = 1, \dots, r$$

## Singular value decomposition

- What about real-valued  $A$ ?
- Consider the case with  $A \in \mathbb{R}^{m \times n}$

$$A = USV^T$$

- Columns of orthogonal  $U \in \mathbb{R}^{m \times m}$ : left singular vectors
- Columns of unitary  $V \in \mathbb{R}^{n \times n}$ : right singular vectors
- Diagonal elements of  $S \in \mathbb{R}^{m \times n}$  are the singular values  
$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0, \quad r = \min\{m, n\}$$
- This is still an SVD because it satisfies the requirements:
  - the first and last matrices are orthogonal,
  - and the middle matrix is diagonal with non negative entries

## Interpreting the SVD

- Let's rewrite the SVD

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^*$$

- As follows (postmultiply by V):

$$\mathbf{A}\mathbf{V} = \mathbf{U}\mathbf{S}$$

- Considering one column at a time, we have

$$\mathbf{A}\mathbf{v}_k = \sigma_k \mathbf{u}_k, \quad k = 1, 2, \dots, r, \quad r = \min\{m, n\}$$

- This means that each right singular vector  $\mathbf{v}_k$  is mapped by  $\mathbf{A}$  to a scalar multiple ( $\sigma_k$ ) of the corresponding left singular vector  $\mathbf{u}_k$

## Interpreting the SVD: example

- Compute an SVD:

```
>> A = [2 1;3 4; 5 6]
A =
    2     1
    3     4
    5     6
>> [U,S,V] = svd(A)
U =
   -0.2164    0.9497   -0.2265
   -0.5258   -0.3088   -0.7926
   -0.8226   -0.0524    0.5661
S =
    9.4939      *    0
        0    0.9303
        0        0
V =
   -0.6450    0.7642
   -0.7642   -0.6450
```

- Compare the first column of  $AV$  with the first column of  $U$ : this should be just  $\sigma_1$

```
>> AV = A*V
AV =
   -2.0541    0.8834
   -4.9917   -0.2872
   -7.8101   -0.0488
>> AV(:,1)../U(:,1)
ans =
    9.4939
    9.4939
    9.4939
>> S(1,1)
ans =
    9.4939
```

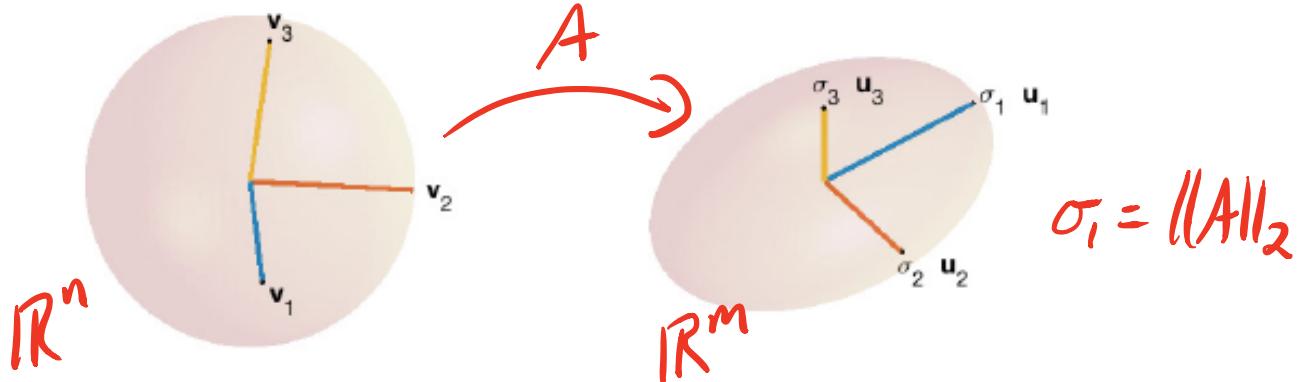
## Interpreting the SVD: visualizing it

- The SVD can be visualized for  $3 \times 3$  real-valued matrices
- Unit vectors in the directions of the columns of  $V$  (left) are distorted by multiplication by  $A$  (right) to

$A_{m \times n}$

$A: \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$\boxed{A} \boxed{\begin{matrix} v \\ u \\ s \end{matrix}} = \boxed{U} \boxed{\begin{matrix} \sigma \\ 0 \end{matrix}}$$



# Contrasting the EVD and the SVD

**Table 7.1.** *Differences between the EVD and the SVD.*

EVD	SVD
most square matrices	all rectangular and square matrices
$\mathbf{A}\mathbf{x}_k = \lambda_k \mathbf{x}_k$	$\mathbf{A}\mathbf{v}_k = \sigma_k \mathbf{u}_k$
same basis for domain and range of $\mathbf{A}$	two orthogonal bases
may have poor conditioning	perfectly conditioned

## Connection between the SVD and 2-norm (Thrm 7.3.2)

- Let  $A \in \mathbb{C}^{m \times n}$  have an SVD with  $A = USV^*$  and  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0, r = \min\{m, n\}$

- Then:

- The 2-norm satisfies  $\|A\|_2 = \sigma_1$ .
- The rank of  $A$  is the number of nonzero singular values.
- The condition number satisfies

$$\kappa(A) = \|A\|_2 \|A^+\|_2 = \sigma_1 / \sigma_r.$$

Division by zero here implies that  $A$  does not have full rank.

Recall that  $A^+ = (A^T A)^{-1} A^T$  is the pseudoinverse (chapter 3).

$$A^+ = V S^+ U^* \quad S^+ = \begin{bmatrix} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & 0 \end{bmatrix}$$

## SVD example 7.3.2: try it

```
➤ A=vander(1:5);      % Vandermonde matrix  
➤ A = A(:,1:4)        % 5 by 4 now  
➤ [U,S,V] = svd(A);  
➤ norm(U'*U-eye(5)) % check U is orthogonal  
➤ norm(V'*V-eye(4)) % check V is orthogonal  
➤ sigma = diag(S)  
➤ [norm(A) sigma(1)]    % Thrm 7.3.2, no. 1  
➤ [cond(A) sigma(1)/sigma(end)] % Thrm 7.3.2, no. 3
```

[Example 7.3.2]

## Connections between SVD and EVD

- Let  $A = USV^*$
- Create the square hermitian matrix  $B = A^*A$
- Then  $B = (VS^*U^*)(USV^*) = VS^*SV^* = V(S^T S)V^{-1}$ .
- $S^T S$  is a diagonal  $n \times n$  matrix
- There are two cases:

$m \geq n$ :

$$S^T S = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix},$$

$m < n$ :

$$S^T S = \begin{bmatrix} \sigma_1^2 & & & \\ & \ddots & & \\ & & \sigma_m^2 & \\ & & & 0 \end{bmatrix},$$

- The lower right zero is  $n - m \times n - m$

$$\sigma(A) = \sqrt{\lambda(A^*A)}$$

## Connections between SVD and EVD

- There are two cases:

$m \geq n$ :

$$S^T S = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix},$$

$m < n$ :

$$S^T S = \begin{bmatrix} \sigma_1^2 & & & \\ & \ddots & & \\ & & \sigma_m^2 & \\ & & & 0 \end{bmatrix},$$

- The squares of the singular values of  $A$  are the eigenvalues of  $B = A^*A$ !
- Conversely, the EVD of  $B$  yields the singular values of  $A$  and the right singular vectors of  $A$
- We could get the left singular vectors from  $AV = US$  by doing one column at a time

## Connections between SVD and EVD

- We could also create

which is  $m + n \times m + n$

for  $A$   $m \times n$

$$C = \begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix}.$$

*A then*

- If  $\sigma$  is a singular value of  $A$ , then  $\sigma$  and  $-\sigma$  are eigenvalues of  $C$
- The associated eigenvector immediately gives a left and right singular vector of  $A$
- This connection is implicitly exploited by software to compute the SVD

## Thin form of the SVD

- Like the QR factorization, we can have both full and thin versions of the factorization
- Let  $A = USV^*$  where  $A$  is  $m \times n$  with  $m > n$
- Then we can write

$$US = [u_1 \ \cdots \ u_n \ u_{n+1} \ \cdots \ u_m] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & & 0 \end{bmatrix}$$

Last  $m - n$  rows are zero here

$$= [u_1 \ \cdots \ u_n] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} = \hat{U} \hat{S},$$

$\hat{U}$  is  $m \times n$ , and  $\hat{S}$  is  $n \times n$ , but still same info about  $A$  !

- Use  $\text{svd}(A, 0)$  to get this in Matlab

## Section 7.4

# Symmetry and definiteness

## Symmetry and Definiteness

- When we had real matrices  $\mathbf{A}$ , we found in chapter 2 that there could be specializations of the factorizations there: e.g.,  $\mathbf{A} = \mathbf{LDL}^T$ , where  $\mathbf{D}$  is diagonal and  $\mathbf{L}$  is lower triangular
- We now study the analogous hermitian case for  $\mathbf{A}^* = \mathbf{A}$
- Let  $\mathbf{A} = \mathbf{USV}^*$  where  $\mathbf{A}$  is  $n \times n$
- Since  $\mathbf{S}$  is real and square,

$$\mathbf{A}^* = \mathbf{VS}^*\mathbf{U}^* = \mathbf{VSU}^*$$

$$\mathbf{S}^* = \mathbf{S}$$

- In this case  $\mathbf{A}$  is diagonalizable with

$$\mathbf{A} = \mathbf{VDV}^{-1} = \mathbf{VDV}^*$$

- Here  $\mathbf{V}$  is unitary, and  $\mathbf{D}$  is diagonal and real
- This means that a hermitian matrix has a complete set of orthonormal eigenvectors (a unitary diagonalization), with real eigenvalues

## Symmetry and Definiteness: example

### Example 7.4.1

The following matrix is not hermitian.

```
A = [0 2; -2 0]
```

```
A =  
     0      2  
    -2      0
```

so it is normal.

```
[V,D] = eig(A);  
norm( V'*V - eye(2) )
```

```
ans =  
2.2204e-16
```

The eigenvalues are pure imaginary.

```
lambda = diag(D)
```

```
lambda =  
0.0000 + 2.0000i  
0.0000 - 2.0000i
```

The singular values are

```
svd(A)
```

```
ans =  
2  
2
```

[Example 7.4.1]

## Symmetry and Definiteness

- Now we have a theorem that says that the condition number for the eigenvalues is bounded above by  $\kappa(V)$
- Here  $V$  is the eigenvector matrix  $\kappa(V) = 1$
- But, it is unitary (or orthogonal), which means that  $\kappa = 1!$
- So, our last theorem implies that the condition number for a Hermitian or normal matrix is one, which is as good as it gets!
- In that case, that is for Hermitian or normal matrices, the eigenvalues can be changed by no more than the norm of the perturbation of the matrix!
- Can we verify this?

## Symmetry and Definiteness: example

### Example 7.4.2

We construct a real symmetric matrix with known eigenvalues by using the QR factorization to produce a random orthogonal set of eigenvectors.

```
n = 30;  
lambda = (1:n)';  
D = diag(lambda);  
[V,R] = qr(randn(n)); % get a random orthogonal V  
A = V*D*V';
```

The orthonormal columns of V become the associated eigenvectors for the eigenvalues in matrix D, which are the same for A. Why?

Eigenvalues are 1,2,...,30 and put on diagonal of matrix D

The qr function takes a random 30 by 30 matrix here, then returns an orthogonal matrix V and upper triangular matrix R

## Symmetry and Definiteness: example

### Example 7.4.2

The condition number of these eigenvalues is one. Thus bounded by the norm of the perturbation to  $A$ .

```
for k = 1:3
    E = randn(n); E = 1e-4*E/norm(E);
    mu = sort(eig(A+E));
    max_change = norm(mu-lambda, inf)
end
```

[Example 7.4.2]

Create perturbation matrix  $E$  that is random and has norm  $1e-4$

```
max_change =
2.5564e-05
max_change =
2.0501e-05
max_change =
2.3712e-05
```

## Symmetry and Definiteness

- This is great, but it is not quite an SVD
- Why? The sign of the diagonal elements could be anything.
- Can we make it into an SVD? Yes!
- The trouble is with  $D$ ; let's rewrite it as a product of two diagonal matrices: one is  $T$  with the sign( $d_{ii}$ ) on the diagonal; the other is  $|D|$ , which has  $|d_{ii}|$  on the diagonal. We still have

$$D = T|D|$$

$$A = USV^*$$

- Substitute to get

$$A = VDV^* = VT|D|V^* = (VT)|D|(V^*)$$

- This *is* an SVD, because the diagonal matrix has nonnegative entries and the left and right matrices are unitary

$$U = VT$$

## Symmetry and definiteness

- Recall the quadratic form  $\mathbf{x}^* \mathbf{A} \mathbf{x}$  where  $\mathbf{A}$  is  $n \times n$  and  $\mathbf{x}$  is  $n \times 1$ .
- For real entries,  $\mathbf{x}^T \mathbf{A} \mathbf{x}$  is the quadratic form of interest
- For  $n=2$ , we can easily write it out:

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = a_{11}x_1^2 + (a_{12} + a_{21})x_1x_2 + a_{22}x_2^2$$

- For some matrices, this quantity remains positive for any nonzero vector  $\mathbf{x}$
- For example, if  $a_{11} > 0$ ,  $a_{22} > 0$ , and  $a_{21} = -a_{12}$ , then the quadratic form reduces to  $a_{11}x_1^2 + a_{22}x_2^2 > 0$  for nonzero  $\mathbf{x}$
- This matrix is called **positive definite**
- If the matrix were symmetric and this were true, then it is symmetric positive definite or SPD

## Symmetry and definiteness

- The SPD matrix is important: it has all positive eigenvalues
- Complex case: a matrix is Hermitian positive definite (HPD) when

$$A = A^* \text{ and } x^* A x > 0$$

for any nonzero compatible  $x$

- It turns out that we can prove some equivalent statements:

### Theorem 7.4.3

If  $A^* = A$ , then the following statements are equivalent.

1.  $A$  is HPD.
2. The eigenvalues of  $A$  are positive numbers.
3. Any unitary EVD of  $A$  is also an SVD of  $A$ .

- This means, e.g., that all positive eigenvalues implies HPD (or SPD)

## Rayleigh Quotient and eigenvalues

- In your linear algebra class, you may have discussed the Rayleigh quotient
- For  $A n \times n$  and  $x n \times 1$  with complex entries, the Rayleigh quotient is

$$R_A(x) = \frac{x^* A x}{x^* x}$$

$$\begin{aligned} v^* A v &= v^* \lambda v \\ &= \lambda v^* v \end{aligned}$$

- In the special case that  $x$  is an eigenvector, say  $v$ , then  $Av = \lambda v$ , and substitution easily gives that  $R_A(v) = \lambda$
- We can conclude that: *the Rayleigh quotient maps an eigenvector into its associated eigenvalue*

## Rayleigh Quotient and eigenvalues

- Consider the Hermitian case  $\mathbf{A} = \mathbf{A}^*$  and the Rayleigh quotient

$$R_A(\mathbf{x}) = \frac{\mathbf{x}^* \mathbf{A} \mathbf{x}}{\mathbf{x}^* \mathbf{x}}$$

- Put in a vector that is close to an eigenvector:  $\mathbf{x} = \mathbf{v} + \epsilon \mathbf{z}$  for  $\epsilon \rightarrow 0$
- Using a multidimensional Taylor expansion gives that

$$R_A(\mathbf{v} + \epsilon \mathbf{z}) = R_A(\mathbf{v}) + 0 + O(\epsilon^2) = \lambda + O(\epsilon^2)$$

- This happens because, for an eigenvector,  $\nabla R_A(\mathbf{v}) = \mathbf{0}$
- This means that if the input vector is within  $O(\epsilon)$  of an eigenvector then the Rayleigh quotient is within  $O(\epsilon^2)$  of the eigenvalue:  $R_A$  does a good job of approximating the eigenvalue! Let's explore this...

# Rayleigh Quotient and eigenvalues

## Example 7.4.3

We construct a symmetric matrix with a known EVD.

```
n = 20;
lambda = (1:n)'; D = diag(lambda);
[V,~] = qr(randn(n)); % get a random orthogonal V
A = V*D*V';
```

The Rayleigh quotient of an eigenvector is its eigenvalue.

```
R = @(x) (x'*A*x)/(x'*x);
format long, R(V(:,7))
```

```
ans =
7.000000000000001
```

# Rayleigh Quotient and eigenvalues

## Example 7.4.3

Now let's try different vectors that are closer and closer to an eigenvector

```
delta = 1./10.^ (1:4)';  
quotient = 0*delta;  
for k = 1:4  
    e = randn(n,1); e = delta(k)*e/norm(e);  
    x = V(:,7)+e;  
    quotient(k) = R(x);  
end  
table(delta,quotient)
```

Every time the input vector gets a factor of 10 closer to the eigenvector, there is a factor of 100 improvement in the eigenvalue approximation (another two zeros after the decimal point)

```
ans =  
      delta          quotient  
-----  
      0.1      7.05738940427937  
     0.01      7.00066684894918  
    0.001      7.00000278235035  
   0.0001      7.00000005557751
```

[Example 7.4.3]

Section 7.5  
Dimension reduction  
(PCA: Principal component analysis)

## Dimension Reduction

- We now return to the SVD
- We want to use it to approximate the information in a matrix with much less storage, or many less numbers
- We will see a few examples of this
- Consider matrix  $A \in \mathbb{R}^{m \times n}$  with  $m > n$  (for now)
- The thin SVD is then

$$A = \hat{U} \hat{S} V^T$$

Diagram illustrating the SVD decomposition:

Matrix  $A$  is  $m \times n$ . It is equal to the product of three matrices:

- $\hat{U}$  is  $m \times m$ .
- $\hat{S}$  is  $m \times n$  and diagonal.
- $V^T$  is  $n \times n$ .

$$US = \hat{A}\hat{S}$$

$$A = \hat{U} \hat{S} V^T = [u_1 \ u_2 \ \dots \ u_n] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix}$$

- We truncated the  $n+1$  to  $m$  columns of  $U$  and rows of  $S$  to get  $\hat{U}, \hat{S}$

## Dimension Reduction

- Now let's rewrite the thin SVD *carefully*
- Note that  $\widehat{U}$  has orthonormal *columns*  $u_i$
- $V$  has orthonormal *rows*  $v_i^T$

$$r = \text{rank}(A)$$

$$\Rightarrow \sigma_{r+1} = \dots = \sigma_n = 0$$

$$\begin{aligned}
 A &= \widehat{U} \widehat{S} V^T = [u_1 \ u_2 \ \dots \ u_n] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix} \\
 &\quad \text{m} \times n \text{ here} \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{n} \times n \text{ here} \\
 &= [\sigma_1 u_1 \ \dots \ \sigma_n u_n] \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix} \\
 &= \sigma_1 u_1 v_1^T + \dots + \sigma_r u_r v_r^T = \sum_{i=1}^r \sigma_i u_i v_i^T,
 \end{aligned}$$

$u_i v_i^T$  is an *outer product* here, with each forming an  $m \times n$  matrix

## Dimension Reduction

- Each of those outer products is weighed with the singular value, then adding them *all* up recovers the original matrix

$$\mathbf{A} = \sum_{i=1}^r \sigma_i u_i v_i^T,$$

- But, it so happens that in many matrices, there are only a few singular values that are sizable, and the rest may be quite small.
- By convention we ordered the singular values:  
 $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$  r ≥ 1
- So, if the largest few singular values are say  $1, 2, \dots, k$ , then maybe we need to keep only the first  $k$  terms to get a good approximation to the content of  $\mathbf{A}$ !
- Let's make this more precise.

## Dimension Reduction

- We can think of the process of keeping more singular values as generating a sequence of matrices for increasing  $k$ , with  $1 \leq k \leq r$
- Then, we only do a partial sum using the first  $k$  terms:

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T = U_k S_k V_k^T.$$

- $U_k$  and  $V_k$  are the first  $k$  columns of  $U$  and  $V$
- $S_k$  is the upper left  $k \times k$  submatrix of  $S$
- Because each  $u_i v_i^T$  is a matrix with unit norm, the overall size of each matrix added is given by the singular value
- Because of this, there are many cases where stopping at a small value of  $k$  relative to  $r$  will give a ~~not~~ very good approx. of  $A$

## Dimension Reduction

$$A_r = A$$

$$\sigma_{r+1} = 0$$

- What do we know about the  $A_k$  ?
- The rank of a sum of matrices is less than or equal to the sum of the ranks of each: thus, the rank of  $A_k$  is at most  $k$
- And, it turns out that  $A_k$  is the *best rank  $k$  approximation* to  $A$ !

### Theorem 7.5.1

Suppose  $A$  has rank  $r$  and let  $A = USV^T$  be an SVD. Let  $A_k$  be as in (7.5.2) for  $1 \leq k < r$ . Then

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T = U_k S_k V_k^T.$$

1.  $\|A - A_k\|_2 = \sigma_{k+1}, \quad k = 1, \dots, r-1.$

2. If the rank of  $B$  is  $k$  or less, then  $\|A - B\|_2 \geq \sigma_{k+1}$ .

## Dimension Reduction: example

- For many matrices,  $k$  need not be very large to get a good approximation to the original matrix
- Example 7.5.1 gives an example with text
- Demo of that example and/or others...

[Example 7.5.1]

[Example 7.5.2]

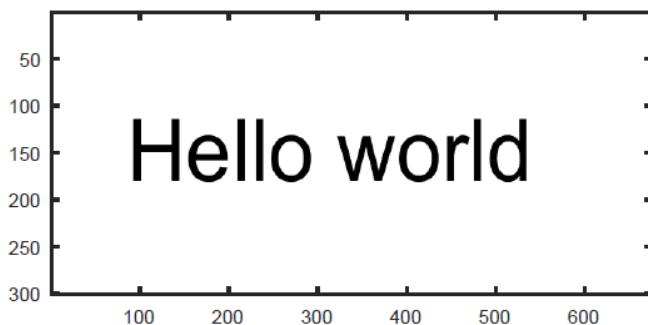
# Dimension Reduction: example

## Example 7.5.1

```
tobj = text(0,0,'Hello world','fontsize',44);
saveas(gcf,'hello.png')
A = imread('hello.png');
A = double(rgb2gray(A));
imagesc(A), colormap gray
[m,n] = size(A)
```

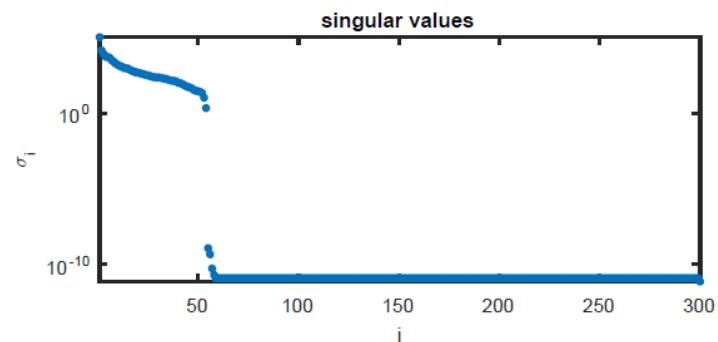
```
m =
  300
n =
  675
```

300x675



```
[U,S,V] = svd(A);
sigma = diag(S);
semilogy(sigma,'.')
r = find(sigma/sigma(1) > 10*eps,1,'last')
```

```
r =
  56
```



## Dimension Reduction: example

Example 7.5.1

```
for i = 1:4
    subplot(2,2,i)
    k = 2*i;
    Ak = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
    imshow(Ak,[0 255])
    title(sprintf('rank = %d',k))
end
```

rank = 2



rank = 4



rank = 6



rank = 8



Look how few singular values are needed to get a decent looking image! Less than 5% of the original storage to get the rank 8 approximation!

## Dimension Reduction: example

- These low rank approximations can be used to get at the essence of data
- One measure of how much content is contained in each added rank is the fractional “energy” given by

$$\tau_k = \frac{\sum_{i=1}^k \sigma_i^2}{\sum_{i=1}^r \sigma_i^2}, \quad k = 1, \dots, r.$$

- Consider a different matrix of information now: Example 7.5.2
- Here we look at the voting pattern for the Senate in the 111<sup>th</sup> session of the US Congress

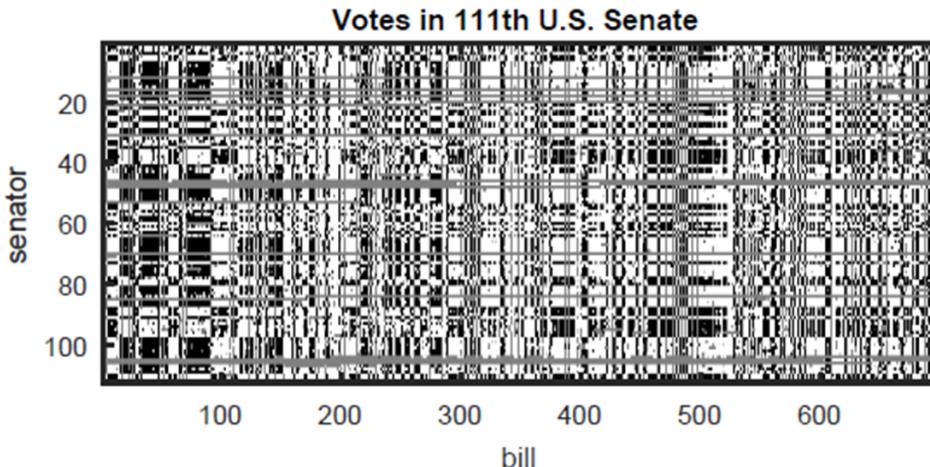
## Dimension Reduction: example

### Example 7.5.2

```
load voting    % get from the book's website  
[m,n] = size(A);
```

If we visualize the votes (white is “yea,” black is “nay,” and gray is anything else), we can see great similarity between many rows, reflecting party unity.

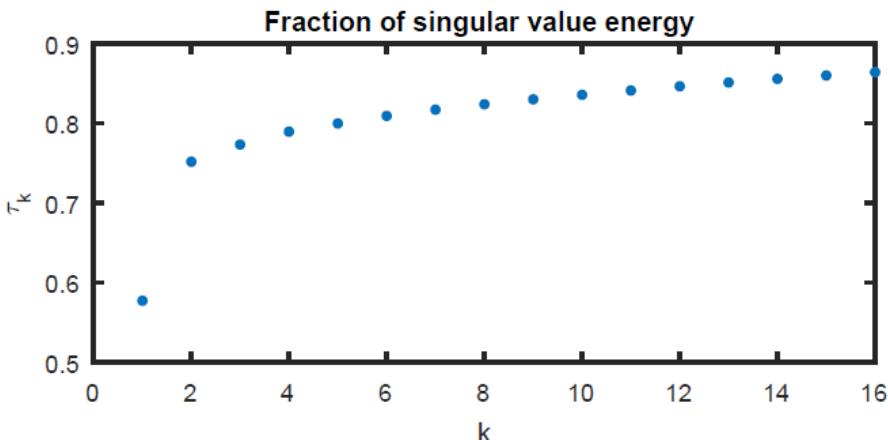
```
imagesc(A)  
colormap gray
```



# Dimension Reduction: example

## Example 7.5.2

```
[U,S,V] = svd(A);  
sigma = diag(S);  
tau = cumsum(sigma.^2) / sum(sigma.^2);  
plot(tau(1:16),'.')
```



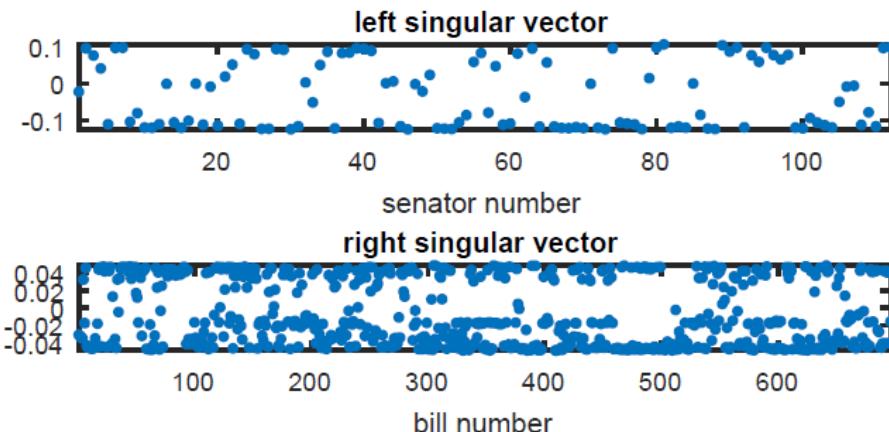
- The first two singular values account for 75% of the energy
- The remaining ones each account for relatively little of the energy
- Maybe just those first two are enough to capture the essence of the data?

## Dimension Reduction: example $A_i = \sigma_i U_i V_i^T$

### Example 7.5.2

```
subplot(211), plot(U(:,1),'.')
subplot(212), plot(V(:,1),'.')
```

$$A_i = \sigma_i [u] [v]^T$$

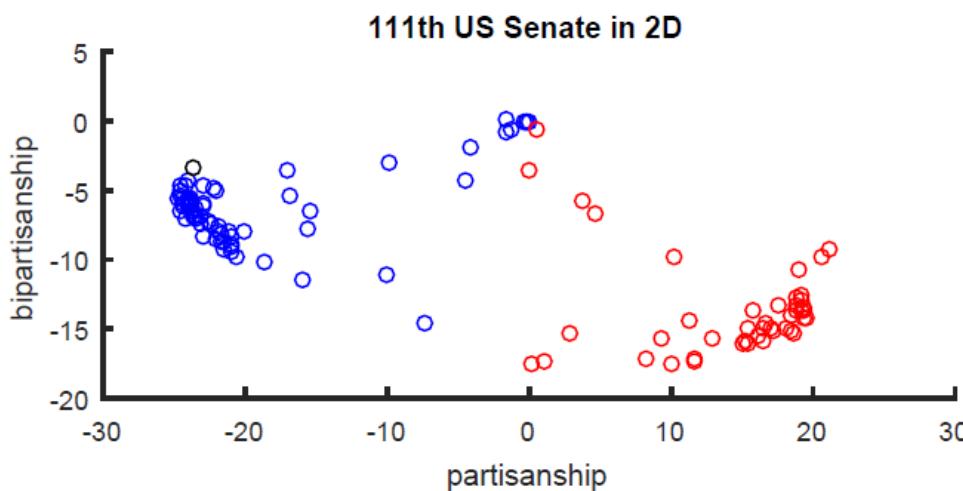


- Note the different sizes of the two vectors
- Most of the values in these vectors are at  $\pm C$ : there is not much in the middle

# Dimension Reduction: example

## Example 7.5.2

```
clf
x1 = V(:,1) '*A';    x2 = V(:,2) '*A';
scatter(x1(Dem),x2(Dem),20, 'b'), hold on
scatter(x1(Rep),x2(Rep),20, 'r')
scatter(x1(Ind),x2(Ind),20, 'k')
title('111th US Senate in 2D')
```



- Projecting each senator's votes in first two **V** coordinates, the right singular vectors: (1) is partisanship; (2) is bipartisanship
- Those coordinates are then plotted against each other
- Red: Republican
- Blue: Democrat
- Black: Independent
- The scatter plot suggests that there is a clear separation between parties!